

1 **Printing web content from self-service kiosks**

2

3 **Field of the Invention**

4

5 The present invention relates to the field of printing web content from self-service kiosks.

6

7 **Background to the invention**

8

9 Self-service kiosks are computing devices placed in the public arena and designed to  
10 provide information, service or products to multiple users in separate self-contained  
11 sessions. Examples of kiosks are automatic teller machines (ATMs), public internet  
12 access terminals, credit card operated pre-paid ticket collection terminals, networked  
13 ticketing machines and computerised vending machines.

14

15 Many self-service kiosks display web content such as web pages written in a markup  
16 language such as HTML, DHTML, XHTML™ or XML. In the course of delivering web  
17 content, kiosks also run applications, including those downloaded across the internet in  
18 languages such as Java® or using component architectures such as ActiveX® or  
19 JavaBeans®. Typically, self-service kiosks that display web content have an internet  
20 connection for downloading the web content from remote servers. However, some or all  
21 of the web content they display can be stored locally, either within the self-service kiosk  
22 or in a commercial intranet or LAN.

23

24 The HTML 4 specification, available from the World Wide Web Consortium  
25 ([www.w3.com](http://www.w3.com)) and other related markup languages, enable a browser window to be  
26 partitioned into multiple frames.

27

28 One common use of frames is to provide graphics and links in a first frame, which do not  
29 change when a new markup language document is retrieved and rendered in a second  
30 frame.

31

1 Furthermore, it is possible for a browser to display more than one window, each of which  
2 may consist of one or more frames. US Patent Applications No. 09/870,293 and  
3 09/870,057 to Aravinda Korala, and to be assigned to Korala Associates Limited disclose  
4 a software architecture for enabling self-service kiosks to display a carefully laid-out web  
5 application using not only different windows implemented by a single browser  
6 component, but windows controlled by a plurality of browser components, each of which  
7 may display one or more browser windows. US Patent Applications No. 09/870,293 and  
8 09/870,057 are hereby incorporated by reference. The ability to operate linked web  
9 commerce applications in multiple frames and windows on screen enables many useful  
10 new types of web commerce transaction to be implemented.

11

12 However, in multiple frame and multiple browser window environments a problem is  
13 commonly faced when a user wishes to make a print instruction. It is desirable for users  
14 to be able to simply select a print command without having to qualify what they want to  
15 print, for example by having to choose options from a dialogue box. For this reason  
16 many browsers, such as Internet Explorér® 5.5 from Microsoft of Redmond, WA,  
17 display a single icon which, when selected, causes the most recently selected frame to be  
18 printed. Options can be changed allowing the print icon to trigger printing of all frames.  
19 However, users often find that the frame which is printed is not that which they wanted or  
20 that, alternatively, more information is printed than they required. Internet Explorer® 5.5  
21 does provide a user with the option to select printing of all windows or just the most  
22 recent window from a dialogue box displayed in response to the PRINT command on the  
23 FILE menu; however, this is not sufficiently flexible for all applications and is  
24 inappropriate on a self-service kiosk in the public arena where menus and dialogue boxes  
25 are typically suppressed. It is also not appropriate to give a user a choice when it is  
26 desired to print a frame which a user cannot see and does not know is present; for  
27 example, when a frame is rendered hidden behind another window and is intended for  
28 printing but not display.

29

30 Therefore, a first aim of the present invention is to enable an appropriate frame or frames  
31 to be printed in response to a simple user request to print.

1  
2 Yet further problems are present when multiple windows are displayed on screen at once.  
3 By default in windowing environments such as Windows®, selecting PRINT in a  
4 browser window will cause the content of one or all frames in that window to be printed  
5 as discussed previously. However, there is no self-service kiosk known to us to have  
6 implemented a satisfactory method for enabling an appropriate frame and/or window or  
7 windows to be printed in response to a simple user request to print when multiple  
8 browser windows or multiple browser components are present.

9  
10 The present invention therefore also aims to enable a simple user print instruction to be  
11 responded to appropriately in a multi-window, multi-frame environment on a self-service  
12 kiosk.

13  
14 A further problem faced with printing from self-service kiosks relates to the particular  
15 printing features available in common operating systems used to operate self-service  
16 kiosks. At the present time most self-service kiosks use a standard operating system such  
17 as Windows NT® 4.0. Most self-service kiosks simply use Windows NT® basic printing  
18 feature; however, this has many limitations, particularly in terms of its ability to report  
19 printer status information to a controlling application. If the printer runs out of paper or  
20 breaks, the kiosk can detect if a print job fails, so it could for example go out of service.  
21 But the kiosk can't give detailed error reports to the controlling application. This makes it  
22 difficult to monitor the self-service kiosk from a remote location as official inspection is  
23 required to check printer status. Neither can the kiosk detect when the printer is running  
24 low of paper or ink, which would allow the kiosk owners sending out an operator to  
25 replenish the paper or ink before the printer fails. .

26  
27 For printing HTML documents, typically the rendering will be done by the browser,  
28 such as the Internet Explorer® 5 browser component, available from Microsoft® of  
29 Redmond, WA. The result of this rendering is a description of the document in the page  
30 layout format of the particular printer. This is then transmitted to the printer by the  
31 appropriate Windows printer driver. The Internet Explorer® 5 print method has some

1 additional useful features, for example the ability to ignore background colour, making  
2 printing of coloured web pages more visible on black and white printers.

3

4 However, this method does not have the useful additional printing features provided in  
5 Windows Extensions for Financial Systems (XFS) available in the XFS specification  
6 available from <http://www.xfsws.com>. For example, XFS print commands provide  
7 paper, load, ink and toner low status information, notification of when printing has taken  
8 place, retraction of paper and more complex paper functions such as cutting, ejecting etc.

9 However,

10 these features are not used with self-service kiosks using Internet Explorer ® 5.5 on a  
11 Windows NT ® operating system.

12

13 A further problem is that printing from self-service kiosks is sometimes charged to the  
14 user per page and it is not immediately possible in Windows ® or in Internet Explorer ®  
15 5.5 to calculate the number of pages which will be required to print an entire page or  
16 frame until the print operation has been completed. This makes it impractical to warn the  
17 user of the cost or prevent a job being started which exceeds the credit available to the  
18 user. Therefore, a further aim of the present invention is to provide new printing methods  
19 to obviate these problems.

20

## 21 **Brief description of the invention**

22

23 According to a first aspect of the present invention there is provided a method of  
24 selecting images for printing on a self-service kiosk, the self-service kiosk having a web  
25 browser for rendering markup language documents and causing said rendered documents  
26 to be displayed on a screen, the self-service kiosk having the capacity to display a  
27 plurality of visual information in a plurality of frames, the contents of each frame being  
28 specified by a different markup language document, the method comprising the steps of:

29

30 receiving a print command from a user; and

31

1       selecting at least one frame for printing dependent on the URL of a displayed  
2       markup language document.

3  
4       This means that a user of the self-service kiosk has only to make a simple print command  
5       and, without having to specify a particular frame or frames, receives a print-out which  
6       has been rationally chosen to consist of relevant selected frames.

7  
8       In further aspects, one or more frames may be selected for printing dependent on  
9       information contained within a displayed markup language document, the location within  
10      a browser window or dimensions of one or more frames or the content of a markup  
11      language document.

12  
13      According to a yet further aspect of the present invention there is provided a method of  
14      printing a markup language document on a self-service kiosk, the method comprising the  
15      steps of:

16  
17       responsive to a print instruction, a markup language rendering module preparing  
18       an output print file containing raw printer data corresponding to said markup  
19       language document;

20  
21       instructing a print module to print the raw printer data in the output print file.

22  
23       This aspect enables a print module capable of providing printer status information, to be  
24       used to provide print output on the kiosk, whilst also benefiting from the rendering  
25       facilities of a markup language rendering module such as a web browser.

26  
27       Preferably, the method further includes the step of counting the number of page breaks in  
28       the output print file and thereby determining a user's bill for printing. Preferably also,  
29       printing will not take place unless the user credit exceeds the printing charge. This  
30       counting of the number of page breaks in the output print file allows calculations and

1 decisions to be made based on the number of pages a document will require to print even  
2 if that document is written in a markup language which does not list page breaks.

3

4 **Description of the several drawings**

5

6 An example embodiment of the present invention will now be illustrated with reference  
7 to the following Figures in which:

8

9 Figure 1 is a schematic diagram of a self-service kiosk;

10

11 Figure 2 is multi-frame browser window;

12

13 Figure 3 is a self-service kiosk screen displaying multiple browser windows with  
14 multiple frames;

15

16 Figure 4 is a flow diagram of the process of the present invention;

17

18 Figure 5 is a block diagram of key components of self-service kiosk control  
19 software;

20

21 Figure 6 is a flow diagram of a print method; and

22

23 Figure 7 is a flow diagram of a print method including page counting.

24

25 **Detailed description of the preferred embodiment**

26

27 Figure 1 illustrates a self-service kiosk 1 having a display 2 and having access to one or  
28 more servers 3 across a network 4. The network 4 is preferably the internet and servers 3  
29 are preferably HTTP servers; however, other networking protocols or internal  
30 communications routes to locally stored information can readily be substituted.

31

1 Kiosks have user interfaces for interaction with users 5 and are controlled by a control  
2 computer 10. In the preferred embodiment, the control computer 10 runs Windows NT®  
3 4.0 but those skilled in the art can select alternative operating system depending on  
4 commercial and programming preferences. User interfaces may incorporate devices such  
5 as monitors, touch screens, keyboards, mice, cash dispensers, card reading devices,  
6 identification devices such as number pads for inputting in a PIN number or iris, retina or  
7 finger print readers. The self-service kiosk 1 has a printer 7 which is used to provide a  
8 hard copy of information. Typically, the information printed will be a copy of  
9 information shown on display 2 within a browser window; however, other information  
10 which has not been displayed will commonly be output, for example a confidential bank  
11 statement.

12

13 Figure 2 shows a web browser window 20 that occupies the whole of a display 2.  
14 Browser window 20 comprising three separate frames 21, 22 and 23. Individual frames  
15 21, 22 and 23 are rendered from separate markup language documents and present  
16 different information. Preferably, frame display documents are rendered by a web  
17 browser and written in a markup language such as HTML, DHTML, XHTML™ or  
18 XML.

19

20 In this example, frame 21 is an advert comprising advertising matter 24 and a link 25 to  
21 an external advert. If a user clicks on the link 25, a separate browser window appears  
22 showing an advert, as is well known in web advertising. A control frame 22 provides  
23 links to various different features pertaining to the service being provided to the user of  
24 the self-service kiosk. Here, the self-service kiosk 1 is delivering a banking application  
25 and a user can display web pages relating to specific accounts by selecting links referring  
26 to their current account 26, savings account 27 or mortgage 28. Clicking on a link 26, 27  
27 or 28 causes a web browser 120 to download and render a different web page in central  
28 frame 23 depending on the link selected.

29

30 In the present example main frame 23 displays a user's account balance 29, options  
31 enabling a user to transfer money 30 or dispense cash 31 and a print option 32, which, if

1 selected, will print a user's balance. If a user wants to print their balance, it is  
2 inappropriate to print either the entire display 2, the advert in frame 21 or control  
3 information in frame 22. It is most appropriate to print the main frame 23 or,  
4 alternatively, a separate document not as yet displayed summarising their account  
5 balance.

6

7 Figure 3 illustrates another example where a user is presented with several different  
8 browser windows 40, 50, 60 and 70 on screen 2. These may be implemented by separate  
9 browser components or can be multiple windows controlled by a single browser  
10 component. Alternatively, the whole structure can be provided using a complex  
11 arrangement of frames in a single browser window. Preferably, separate windows 40, 50,  
12 60 and 70 are provided. Control window 40 provides a central menu enabling a user 5 to  
13 book a holiday.

14

15 Hotel booking window 50 enables a user to select a hotel. Link frame 51 contains  
16 different icons for different hotels and as a user selects different icons 52 from link frame  
17 51, different hotel pictures 53 and additional information about the selected hotel 54 are  
18 shown in the main frame 55 of window 50.

19

20 A separate flight booking window 60 enables a user 5 to select flights for a holiday.  
21 Flights provided by different carriers can be selected from a menu provided in link frame  
22 61 and information for a particular airline is displayed in frame 62. Conventional on-line  
23 air flight booking technology enables a user to select and display information about a  
24 flight.

25

26 Car booking window 70 enables a user to book car hire, selecting different models to  
27 view from link frame 71 which determines which set of information is displayed in frame  
28 72. Additional advertising is shown in frame 73. The use of multiple browser windows  
29 enables a user 5 to interact separately with hotel, car and flight booking applications.

30

1 Control window 40 presents a user with options including confirming the holiday 41 or  
2 printed a summary of a holiday that they designed by selecting a hotel, flights and a car in  
3 the other windows. A user may only confirm the holiday when appropriate hotels, flights  
4 and car hire have been selected. However, if a user selects print summary 42 by selecting  
5 the print summary button 42, the summary that is printed should not be any one of the  
6 browser windows, nor indeed the whole screen 2. In the present example, individual  
7 frames would be selected and displayed, for example frame 55, frame 62 and frame 72  
8 are printed in turn. A separate HTML document is prepared in a hidden window (not  
9 shown) containing frames 55, 62 and 72 and then printed after information that has not  
10 been shown on screen has been added to it.

11

12 Figure 4 is a flow chart illustrating the print process of the present invention. Initially, a  
13 print command is received from a user 80 or from an application. Thereafter, print  
14 handling module 110 determines which frames to print 81 and then causes said selected  
15 frames to be printed 82. This can be done by traversing the Document Object Model  
16 (DOM) of the document until the correct frame is found, and then passing this to the  
17 Windows Object Linking and Embedding (OLE) Application Programming Interface  
18 (API) command “Exec” with the parameter OLECMDID\_PRINT.

19

20 Figure 5 shows in schematic form the software architecture of the preferred embodiment  
21 of the present invention. A container application 100 comprises a print handling module  
22 110 and one or more browser components 120 for downloading and displaying web  
23 pages 130. Each browser component 120 may render the contents of more than one  
24 browser window and each window may have one or more frames. Preferably, rules 115  
25 are provided for determining whether a page should be printed.

26

27 The print procedure can be initialised in different ways. Preferably, a print command will  
28 be generated in response to a user selecting an icon 42 in a displayed web page which  
29 generates an event and the print handling module 110 is responsive to that event. A web  
30 page written in a language such as HMTL, DHTML, XHTML™ or XML™ can call a  
31 software method responsive to the event. In this embodiment, a web page 130 has been

1 retrieved by a browser 120 and specifies a user interface including a visual element  
2 which, when selected, calls the software method print handling module 110.

3

4 A print command may also be generated by a user pressing a dedicated button on a self-  
5 service keyboard, by clicking on a 'print' button in a separate area of the screen from the  
6 browser windows, or automatically by an application. For example, an application  
7 running on the self-service kiosk may directly call print handling module 110.

8

9 Thereafter, print handling module 110 determines 81 which frame or frames should be  
10 printed. Preferably, this is dependent on one or more print determining rules 115. Print  
11 determining rules may be provided as program code or simply stated in a file which is  
12 parsed and interpreted by the print handling module 110.

13

14 Print determining rules 115 may include rules conditional on the URL of a web page.  
15 For instance, when a print handling module 110 is called by a particular web page 130,  
16 the print handling module 110 may compare a URL with a stored list of URLs 116  
17 defined individual URLs or sets of URLs and lists which frame should be printed for  
18 each URL specified in the rules.

19

20 In a further sophistication, rules 115 may take into accounts the URLs of individual  
21 frames. For example, rules may specify that certain information at certain URLs should  
22 be printed if those URLs are displayed in any frame.

23

24 Rules 115 may also be conditional on information contained in web pages, for example  
25 web pages may contain meta tags associated with this print feature. Print handling  
26 module 110 can be readily adapted to search through each web page currently being  
27 displayed for all documents comprising the meta tag and then print each frame containing  
28 the meta tag.

29

30 Rules 115 may also be conditional on the location of a frame on screen, for example in  
31 the case of figure 2, frame 21 covers the entire width of the screen and is wider than it is

1 tall. This location and height to width ratio are commonly associated with adverts which  
2 preferably would not be printed.. Similarly narrow windows on the left hand side of the  
3 screen are also often not appropriate to print 22. In this case, rules 115 can specify  
4 algorithms identifying inappropriate frame by their location and dimensions. In this case,  
5 the location and height to width ratio of frames 21 and 22 indicated that the only other  
6 frame, 23, should be printed.

7  
8 As a result, a user can activate a simple print command without having to make a  
9 complex selection from a dialogue box and without having to choose from several  
10 different print options. The resulting hard copy is rationally chosen to be appropriate to  
11 the web commerce application that users have been using.

12  
13 In a further embodiment there are provided sets of rules in a format referred to herein as  
14 sitesets. Sitesets specify information relating to individual services provided on a self  
15 service kiosk. For example, one siteset may define information pertaining to the banking  
16 application shown in Figure 2. A siteset preferably defines one or more URLs of initial  
17 web pages to display when a particular self-service kiosk function is selected. Sitesets  
18 may also specify URLs which a user should not be allowed to access. Sitesets may  
19 specify the location and dimensions of browser windows 40, 50, 60 and 70 on the display  
20 2 and may additionally specify a URL of a web page to which the self-service kiosk  
21 should revert at the end of a session with an individual user. Further information about  
22 sitesets is provided in US Application 09/870,057 to Aravinda Korala, and to be assigned  
23 to Korala Associates Limited and incorporated herein by way of this reference.

24  
25 In a further embodiment of the present invention there are provided one or more  
26 selectable agent software applications, each may be activated or de-activated in response  
27 to a user selection. An individual site agent defines functionality which is called when  
28 individual web pages 130 are downloaded by browser modules 120. Site agent modules  
29 may for example, send instructions to peripheral devices to carry out particular functions,  
30 they might send instructions to and receive information from web pages, site agents  
31 receive events whenever web pages 130 are downloaded whilst they are activated. The

1 print handling module 110 may be part of the site agent, so that the site agent contains the  
2 logic for determining which frame or frames are to be printed. Further information about  
3 site agents is provided in US Application 09/870,293 in the name of Aravinda Korala, to  
4 be assigned to Korala Associates Limited.

5

6 The invention further provides an improved method of printing on a self-service kiosk  
7 once the particular markup language documents to be printed have been established.  
8 Figure 6 is a flow chart of a printing method. The method begins 200 when the frame,  
9 frames or other information to be printed have been determined. Firstly, a web browser  
10 120 is instructed to render and print 210 a mark-up language document 215 to a file 216.  
11 This may be carried out by passing this frame, frames or other information to a browser  
12 component such as Microsoft® Internet Explorer® using the Windows OLE API  
13 command "Exec" with the parameter OLECMDID\_PRINT.

14

15

16 Internet Explorer ® browser component outputs raw printer data to file 216. Thereafter  
17 the information is sent 220 to the printer using a self-service kiosk print module.  
18 Preferably, the self-service print module is Kalignite® KXStatementPrinter ActiveX®  
19 control available from Korala Associated Limited of Edinburgh, Scotland or is  
20 Kalignite® KXStatementPrinter Javabean available from Korala Associated Limited of  
21 Edinburgh, Scotland or is an OPOS™ print control object such as the OPOS 1.6 Printer  
22 common control object available Monroe Consulting Services, Dayton, Ohio, XFS  
23 printer service provider such as Data Techno BDT service provider available from Korala  
24 Associated Limited of Edinburgh, Scotland or JXFS com.jxfs.service.IjxfsPrinterService  
25 Java class available from IBM of White Plains, New York. Each of these modules  
26 provides control over self-service features such as stacking and ejecting printout and  
27 capturing untaken printout. Each of these modules also provides status information such  
28 as low paper that exceeds the status information available from the standard facility  
29 provided by Windows NT 4.0®.

30

1 In one embodiment, data 216 for sending to a printer is raw printer data in appropriate  
2 format such as Postscript®. In an alternative embodiment the browser is instructed to  
3 output data 216 in the form of a bit map. A bit map can then be embedded into a XFS  
4 form which contains a single field of type bit map. This form is then printed by the XFS  
5 print control.

6

7 In a further alternative embodiment a container application implements its own renderer  
8 which converts HTML into the XFS form language not as a bit map but as a series of  
9 appropriate fields (text, graphics, bit map etc) for insertion into a form. Importantly, this  
10 means that rendered information from an HTML document or other similar mark-up  
11 language document, such as DHTML, XHTML™ or XML can then be printed using  
12 XFS.

13

14 Figure 7 illustrates a preferred embodiment including the additional step of counting 217  
15 the number of pages in data file 216. This is now possible because raw printer data will  
16 comprise page breaks which can be readily identified by scanning through data file 216.  
17 Page breaks are, however, not provided in the initial mark-up language document 215  
18 and so the number of pages could not be correctly determined before rendering 210.  
19 After counting the number of pages, a decision 208 can then be made as to whether or not  
20 to allow printing the document. For example, the cost of printing the required number of  
21 pages can be compared to the amount of credit a user has in which case the procedure  
22 either continues 220 or is alternatively cancelled if insufficient credit remains. Even if  
23 there is no set credit limit, counting the number of pages 217 enables a user to be billed  
24 appropriately depending on the number of pages that have been printed. Alternatively the  
25 kiosk could display the cost to the user and display a message asking the user to confirm  
26 the print request. This method can also help track the number of sheets of paper left in  
27 the self-service kiosk 1.

28

29 This method therefore allows the benefits of a web browser mark-up language renderer to  
30 be combined with the improved feedback provided in self-service kiosk print controls.

31

1 The invention extends to computer programs in the form of source code, object code,  
2 code intermediate sources and object code (such as in a partially compiled form), or in  
3 any other form suitable for use in the implementation of the invention. Computer  
4 programs may be standalone applications, software components, scripts or plug-ins to  
5 other applications. Computer programs embedding the invention may be embodied on a  
6 carrier, being any entity or device capable of carrying the computer program: for  
7 example, a storage medium such as ROM or RAM, optical recording media such as CD-  
8 ROM or magnetic recording media such as floppy discs. The carrier may be a  
9 transmissible carrier such as an electrical or optical signal conveyed by electrical or  
10 optical cable, or by radio or other means. Computer programs may be provided for  
11 download across the internet from a server. Computer programs may also be embedded  
12 in an integrated circuit. Any and all such embodiments containing code that will cause a  
13 computer to perform substantially the invention principles as described, will fall within  
14 the scope of the invention.

15  
16 Further modifications and alterations can be made within the scope of the invention  
17 herein disclosed.

18